# Chapter 5 System Functionality

## 5.1 Authentication

Authentication refers to the process of verifying who a user is and subsequently granting him access to a system. A safe authentication system was developed in this project with Supabase Authentication and JWT (JSON Web Tokens) to manage the sessions. Users make an account using a username and a password. To provide more security, the password is hashed and then stored in the database that is, it is stored in a scrambled format which cannot be undone to save user credentials even when the database is retrieved. There are two types of users:

* The regular users are allowed to register freely with their email and password.
* Organizations are able to use a distinct Organization ID to access. The system administrators only issue this ID to known groups, and they are officially registered.

Although there was no implementation of traditional session mechanism (such as log out timer), after the user logs in, the system will remember the session of a user even between page loads, and therefore a user is not automatically logged out even when navigating or refreshing the page.

This method ensures that:

* Identities of users are authenticated.
* Sensitive data is protected.
* Only authorized users (e.g., verified organizations) can access certain features.

## 5.2 User Data Storage

Once a user is logged in successfully, the system will monitor the activity and progress of the user. This is carried out by saving user information in the back end with Supabase which is a secure online database. Each user has the following information saved:

* Name and email address
* Hashed password (to be secure)
* Points earned
* Time and date of registration
* Every sustainability activity done.
* Donation history
* Login timestamps

This information is applicable in two ways:

1. Personal use -The user is able to view their status, points, completed actions and donor counts on the dashboard.
2. Organization reporting - The same data is displayed (in summary form) in the organization administration dashboard. This aids organizations to know the number of people using it, the activities that are popular and people who are doing a good job.

Also, the leaderboard option runs on this stored information. It ranks all of them according to the number of total points and shows the ranking, which brings friendly competition, and makes users want to participate more. In general, this user data storage system will guarantee that:

* Every development is stored and presented properly.
* Organizations possess defined understanding of user engagement.
* Users are able to see and monitor their sustainability experience in real time.

## 5.3 Donation System

The donation system enables the user to contribute to sustainability causes using actual money. It is created with Stripe, a safe and popular online payment system. Here’s how it works:

**Integration with Stripe:** A payment form powered by Stripe allows users to donate with their card. It is a safe and easy procedure.

**What Data Is Stored:** A user can make a donation where the following information is stored safely in the database:

* Amount donated
* Time and date of donation
* User ID/email (to trace who is donating)

Note: Card information is not saved to provide privacy and security of users.

**User Profile Updates:** Once the user has donated, a profile displays the amount of donation. This will make users maintain their records on what they contribute and take more action.

**Impact on Points and Rewards:** Donations are also given more points than the normal activities. These points:

* Improve user rankings on the leaderboard
* Unlock better rewards
* Increase their sustainability badge tier

Such a system encourages users to repay, and, at the same time, provides organizations with useful data to measure engagement and impact via the admin dashboard.

## 5.4 Q-Learning AI Recommendation System

To enhance user engagement and personalize the experience, a **Q-learning-based AI recommendation system** was integrated into the P.A.C.E. platform. This smart system assists the users in being directed to most efficient sustainable steps depending on their past activity, making the participation simpler, more influential and more rewarding.

**What is Q-Learning?**

Q-learning represents a form of reinforcement learning algorithm, in which the system is reinforced to understand which actions are optimal to take under particular circumstances as time passes. It does not need a model of the environment but rather depends on experiences gained by means of interactions.

Every user action in the P.A.C.E. system is considered a potential choice (e.g., donating, recycling, volunteering). The system stores these actions and their outcomes in a Q-table, where:

* **Rows** represent user states or situations. Each action has 2 activities each(e.g. if in Action\_1 user already did an Activity\_1/Activity\_2 then the state flag is set to 1 )
* Columns represent available actions(e.g. see above- if Activity\_1/Activity\_2 is set as flag 1 then it is considered as that Action\_1 is completed, so flag 1 for Action\_1)
* The value in each cell in q-value denotes the number based on the below Q-algorithm formulae where it calculates based on most performed activity.

When the user interacts with the system, the Q-values are updated by the Bellman Equation:

Q(s,a)=Q(s,a)+α⋅[r+γ⋅maxQ(s′,a′)−Q(s,a)]

Where:

* s = current state,
* a = action taken,
* r = reward received,
* s′ = new state,
* alpha-α = learning rate,
* gamma-γ = discount factor.

**How the Q-Table Was Trained**

The process of the training was carried out in four significant steps:

Step 1: Initial Setup

* All possible states (e.g. low points, medium points, high points) and all possible actions (Recycle, Donate, Volunteer, etc.) were specified.
* A reward matrix was constructed -rewards on more significant action such as donation being assigned more rewards.

Step 2: Exploration

* The model started experimenting on the various actions in each state.
* It studied the extent to which action received a reward (points), under various circumstances.

Step 3: Learning

* Based on the reward obtained and the future reward that would be obtained, the Q-values (scores) were updated after every action using the Bellman Equation.
* The model was run through numerous cycles (referred to as episodes) and therefore had the opportunity to learn through several tries.

Step 4: Final Q-Table Generation

* A stable Q-table was produced after training. This table assists the system in determining what is the best step to take of a user according to his/her current point level.

It was implemented in Python and the resulting Q-table was saved in the Supabase database.

**Frontend Integration (Website UI)**

The proposal is present on the screen as a unique user interface:

* A bracket wraps around the most recommended activity.
* This bracket dynamically moves according to the way Q-table recommends it to.
* This visual signal is the guide of AI that helps the user pay attention to the most effective activity to get more points.

**Backend Integration**

* Since it is a background service, the Q-learning code is written in Python.
* It connects to Supabase database to access the activity logs and update the Q-table.
* The react frontend fetches the recommendation after its calculation and presents it through the bracket UI around the activity boxes.

**Impact and Benefits**

* It ensure that users do not need to second guess the next action to take- they have a clear call of action.
* The AI bases its suggestions on actual information, and this increases trust and makes people interested.
* It helps users **earn points faster**, and thus users can move up the leaderboard and access higher reward tiers much faster.This Q-learning solution achieves platform targets